Inf Retrieval (2006) 9:435–453
DOI 10.1007/s10791-006-6391-7

Can constrained relevance feedback and display strategies help users retrieve items on mobile devices?*

Vishwa Vinay· Ingemar J. Cox· Natasa Milic-Frayling· Ken Wood

Received: 1 July 2005 / Revised: 17 November 2005 / Accepted: 29 November 2005

Abstract Searching online information resources using mobile devices is affected by small screens which can display only a fraction of ranked search results. In this paper we investigate whether the search effort can be reduced by means of a simple user feedback: for a screenful of search results the user is encouraged to indicate a single most relevant document. In our approach we exploit the fact that, for small display sizes and limited user actions, we can construct a user decision tree representing all possible outcomes of the user interaction with the system. Examining the trees we can compute an upper limit on relevance feedback performance. In this study we consider three standard feedback algorithms: Rocchio, Robertson/Sparck-Jones (RSJ) and a Bayesian algorithm. We evaluate them in conjunction with two strategies for presenting search results: a document ranking that attempts to maximize information gain from the user's choices and the top-D ranked documents. Experimental results indicate that for RSJ feedback which involves an explicit feature selection policy, the greedy top-D display is more appropriate. For the other two algorithms, the exploratory display that maximizes information gain produces better results. We conducted a user study to compare the performance of the relevance feedback methods with real users and compare the results with the findings from the tree analysis. This comparison between the simulations and

* Extended version of "Evaluating Relevance Feedback Algorithms for Searching on Small Displays," Vishwa Vinay, Ingemar J. Cox, Natasa Milic-Frayling, Ken Wood published in the proceedings of ECIR 2005, David E. Losada, Juan M. Fernández-Luna (Eds.), Springer 2005, ISBN 3-540-25295-9

V. Vinay (✉) · I. J. Cox
Department of Computer Science, University College London, UK
e-mail: v.vinay@cs.ucl.ac.uk

I. J. Cox
e-mail: ingemar@ieee.org

N. Milic-Frayling · K. Wood
Microsoft Research Ltd., 7 J.J. Thomson Avenue, Cambridge, UK
e-mail: natasamf@microsoft.com

K. Wood
e-mail: krw@microsoft.com

real user behaviour indicates that the Bayesian algorithm, coupled with the sampled display, is the most effective.

Keywords Relevance feedback · Display strategies · Small displays

1. Introduction

The continuing evolution of portable computing and communications devices, such as cell phones and Personal Digital Assistants (PDAs), means that more and more people are accessing information and services on the Internet with devices that have small displays. This small display size presents challenges. First, a need for extensive scrolling makes viewing of standard pages very difficult. Second, the input modes on PDAs or mobile phones are far less efficient than keyboard typing and make even a simple task of sending a text query rather time consuming. Finally, devices like mobile phones still lack computing resources and speed to perform sophisticated processing on the client side.

We are particularly concerned with the implications that small display devices have on searching online information resources. Generally, it has been observed that users engage in a variety of information seeking tasks, from "finding" a specific, well defined piece of information, to "gathering information" as a more open ended, research oriented activity (Rodden et al., 2003). Use of Internet enabled mobile phones is still in its infancy and no general patterns of use have been established. Anticipating that mobile users will search for specific, well-defined information, we are interested in understanding how relevance feedback, display strategies, and other interactive capabilities can support users engaged in searching for a target document or piece of information.

In this study we explore the effectiveness of three relevance feedback methods in assisting the user to access a predefined target document through searching or browsing. To study this problem, we devised an innovative approach which exploits the fact that the display is small in size and the user's choices are therefore limited. It is then feasible to generate and study the complete space of the user's interactions and obtain an upper bound on the effectiveness of the applied relevance feedback. This bound represents the actions of an "ideal user" who at every step makes choices that enable the system to reach the target document in the minimum number of iterations.

We believe that analysis of the complete search space is a novel experimental paradigm and can lead to interesting insights into the behavior of relevance feedback algorithms. This approach has the further advantage of permitting us to study relevance feedback and display strategies without undertaking time-consuming user studies. It allows us to perform a large number of experiments and collect statistics that could be used to predict the actual user performance. This is demonstrated in our user study described in Section 8.

In Section 2 we give an overview of the related research for mobile devices and relevance feedback and describe the particular algorithms we explore. In Section 3 we describe two display strategies for presenting search results: (i) the display that maximizes the likelihood that the target is in the display (Top-D), i.e., the top ranked documents supplied by the search engine, and (ii) the display that maximizes the immediate information gain from the user's feedback, i.e., selection of relevant documents. Experimental results further characterize these two strategies. In Section 4 to 6 we describe the experimental procedure and discuss the representation and analysis of the user's interaction space. In Section 7, we use these results to construct a compact representation of the statistical model of our simulated users. We validate the approach and the derived user models by way of a user trial, described in

Section 8. We conclude with a summary of the presented work and an outline of the future research directions.

2. Background

A considerable body of research has been dedicated to the issues related to user interaction (Jones et al., 1999; Jones and Marsden, 1997), browsing (Buyukkokten et al., 2000; Buyukkokten et al., 2010), searching (Rodden et al., 2003; Sellen et al., 2002), and reading (Chen et al., 2003) on mobile devices. The idea of using relevance feedback or other adaptive methods to aid searching is not new.

Most directly relevant to our study is Toogle (Ruvini, 2003), a front-end desktop application that post-processes Google results based on the user's actions. Toogle collects evidence that the presented documents are relevant or non-relevant documents from the user's clicks on documents in one or more screens of search results. It uses this information and machine learning techniques to re-rank the remaining documents. In contrast, our approach focuses on searching using mobile devices and constrains the user feedback to the selection of a single relevant document from a small number of documents presented at each iteration.

In our approach we take advantage of the small display size and limited user's actions to study the full space of the user's interactions and all possible outcomes determined by the relevance feedback and display strategies. We are thus able to identify as part of our simulation the 'ideal' user's actions and provide an upper bound on the performance of relevance feedback systems for small displays.

There are several research efforts that share some aspects of our approach. The interactive nature of the task makes it similar to the Ostensive Retrieval Model (Campbell and van Rijsbergen, 1996), except that we are interested in standard relevance feedback algorithms. Very recently, White et al. (2004) measured the performance of implicit feedback models by conducting a simulation-based evaluation. With regards to the experimental setup, our methodology is also similar to that used in Magennis and van Rijsbergen (1997).

The use of a single document as feedback, which the system then uses to automatically infer a new ranking over the data collection, has been previously studied by Aalbersberg (1992). The motivation of providing the user with a manageable interface while taking advantage of relevance feedback stands in our case too. In this paper, we propose an evaluation framework which extends Aalbersberg's use of a single document in the display to one where we provide the user with multiple items in every iteration, but expect only binary feedback regarding the relevance of one chosen document from the displayed set. The effect of using *non-matching* documents for feedback has been shown by Dunlop (1997) and our probabilistic sampled display update provides a semi-principled and computationally efficient method for achieving this end.

2.1. Relevance feedback

Conceptually, a system that involves user relevance feedback can be described by an iterative process. During a display phase, typically a list of documents, the user is given an opportunity to indicate which documents are relevant and which are not. This information is then used by the relevance feedback algorithm to induce a new ranking of documents in the database. The new ranking is the basis of the next display of documents to the user. And the process repeats. The process may begin with an initial query sent to the ranking engine or by a selection of

documents generated by the system itself. A good overview of relevance feedback techniques can be found in Harman (1992).

In our case, the display is a selection of four documents from the ranked list. The user feedback phase is a single action where the user nominates one of the four displayed documents as most relevant to his or her information need. The document ranking phase applies one of three relevance feedback algorithms, described below, to induce the next ranking over the document collection.

2.2. The Rocchio algorithm

The Rocchio relevance feedback scheme (Rocchio, 1971) is used in conjunction with the term-frequency inverse-document-frequency (tf-idf) representation where documents and queries are represented as vectors of term weights and similarity is measured by the cosine dot product between these vectors.

A document is a vector $\boldsymbol{d}_i = (d_{i,1}, d_{i,2}, \ldots, d_{i,W})$ where W is the number of words across the collection, excluding a predefined set of stopwords, and $d_{i,j} = t(i,j) \cdot s_j$. Here $t(i,j)$ corresponds to the number of occurrences of term j in document i and s_j is the inverse document frequency of term j across the whole collection. A query $\boldsymbol{q} = (q_1, q_2, \ldots, q_W)$ is defined similarly, though their values are typically 0 or 1. Both documents and queries are normalized for length by setting

$$d' = \frac{d}{\|d\|} \quad \text{and} \quad q' = \frac{q}{\|q\|} \quad \text{where } \|x\| = \sqrt{\sum_{j=1}^{W} x_j^2}$$

and the similarity score between document $\boldsymbol{d}$ and query $\boldsymbol{q}$ is then given by the dot product of the normalized vectors, i.e., $score_{rocchio}(\boldsymbol{d}_i, \boldsymbol{q}) = \boldsymbol{d}_i' \, \boldsymbol{q}'$. The Rocchio algorithm takes a set $\mathbf{R}$ of relevant documents and a set $\mathbf{N}$ of non-relevant documents (as selected in the user feedback phase) and updates the query weights according to the following equation:

$$w_j' = \alpha w_j + \beta \frac{\sum_{i \in \mathbf{R}} d_{i,j}}{n_{\mathbf{R}}} + \gamma \frac{\sum_{i \in \mathbf{N}} d_{i,j}}{n_{\mathbf{N}}}$$

where n_R and n_N are the number of relevant and non-relevant documents respectively. We use $\alpha = \beta = 1$, and since we do not have non-relevant documents we have $\gamma = 0$.

2.3. The Robertson/Sparck-Jones algorithm

In the Robertson/Sparck Jones model of information retrieval (Robertson and Sparck-Jones, 1976), the terms in a corpus are all assigned relevance weights which are updated for a particular query whenever relevant documents are identified. Initially the relevance weights are given idf-based values. Documents are given ranking scores against a query based on the relevance weights of the query terms occurring in each document. We use the following formulation of this model. The initial relevance weight for term j is given by

$$w_j = \log(C/n_j)$$

where C is the total number of documents in the corpus and n_j is the number of documents containing term j. A document $\boldsymbol{d}_i$ is assigned a score against query $\boldsymbol{q}$ as follows:

$$score_{rsj}(d_i, q) = \sum_{j \in Q} \frac{(K+1) * t(i,j)}{K(1-b) + \frac{b*|d_i|}{l} + t(i,j)}$$

where $t(i, j)$ is the number of occurrences of term j in document $\boldsymbol{d}_i$ with the document length $|\boldsymbol{d}_i|$. K and b are parameters typically set to 2.0 and 0.75, respectively, and l is the average length of all documents in the corpus.

Documents are then ranked in descending score order. If certain documents are flagged as relevant, the relevance weights are updated as follows:

$$w_j = \log\left(\left(\frac{(r_j + 0.5)}{(n_j - r_j + 0.5)}\right)\left(\frac{(C - n_j - n_R + r_j + 0.5)}{(n_R - r_j + 0.5)}\right)\right)$$

where w_j is the weight for term j, n_R is the number of relevant documents and r_j is the number of relevant documents containing term j. C and n_j are defined as before.

In addition to updating the relevance weights, the relevant documents are used to select new (or additional) query terms according to the offer weights, o_j , where $o_j = r_j * w_j$ Terms are ranked in decreasing order of offer weight, and the top terms are used as part of the subsequent query. How many such terms are to be chosen per iteration is another parameter of the system. Choosing this number is problematic in our case. Based on limited evidence (a single relevant document), if a large number of terms is appended to the query at every iteration, the query becomes very noisy. On the other hand, picking only a small number could lead to very discriminatory terms being picked (i.e.; those that are present only in the relevant document). We achieved best performance when expanding the query by a single term in each iteration.

2.4. The Bayesian algorithm

The Bayesian relevance feedback algorithm (Cox et al., 2000), first proposed for a Content-Based Image Retrieval System—PicHunter—is a recursive probabilistic formulation in which, at each iteration, k, the probability, P_k of document $\boldsymbol{d}_i$, being the target document, $\boldsymbol{d}_T$, is computed. This probability is conditioned on all current and past user actions and the history of displayed documents, which collectively is denoted by H_k. The concept of a current query, $\boldsymbol{q}$, is not explicitly present in this formulation. Thus, in each iteration, the document rankings are given by

$$score_{bayesian}(d_i) = P_k(d_i = d_T \mid H_k) = P_{k-1}(d_i = d_T \mid H_{k-1}) * G(d_i, R))$$

where P_{k-1} is the document's probability in the previous iteration and R is the set of documents marked relevant in this iteration. The term $G(\boldsymbol{d}_i, R)$ is given by

$$G(d_i, R) = \prod_{j \in R}\left(\frac{\exp\left(\frac{\text{sim}(d_i,d_j)}{\sigma}\right)}{\left(\sum_{((k \in D) \text{ and } (k \notin R))} \exp\left(\frac{\text{sim}(d_i,d_k)}{\sigma}\right)\right) + \exp\left(\frac{\text{sim}(d_i,d_j)}{\sigma}\right)}\right)$$

The term sim(x, y) computes the similarity of document $\boldsymbol{x}$ with document $\boldsymbol{y}$, which for textual documents can be taken as the cosine dot product of tf-idf vectors normalized for length. The variable σ is a tuning noise parameter which is set according to the specific dataset.

3. Display strategies

At each iteration, it is necessary to display D documents to the user. The most obvious strategy is to display D documents with the highest rank. After successive query refinements (i.e. multiple iterations of feedback), this *Top-D display* is likely to result in a set of documents very similar to one another. If these documents are also similar to the target document or even include it, then this may well be optimum. However, if they are not similar to the target document, the user relevance feedback is unlikely to help redirect the search away from the displayed documents and towards the target.

This problem has been previously discussed in the context of content-based image retrieval (Cox et al., 2000) and observed in the current experiments (see Section 6.2 on Convergence). An alternative approach is to display documents for which a user's response would be most informative to the system and used to minimize the number of search iterations. This was proposed by Cox et al. (2000) and formulated as a problem of finding a selection of D documents that maximizes the *immediate information gain* from the user's response in each iteration. Determining such a document selection is computationally expensive. However, it can be approximated by sampling D documents from the underlying similarity score distribution using computationally efficient methods.

For example, the sampling method may simulate a roulette wheel with the size of each item's field proportional to the relevance score of a document with respect to the specific query. Within such *sampled displays* both documents with high and low ranking have a non-zero probability of being included. Thus the display exhibits more variability and enables the user to direct the search away from a local maximum. We expect the sampled display strategy to be useful in situations where the initial query is imprecise, i.e., when the target document is ranked very low in the search result list.

Using devices with small displays for search thus raises issues similar to those encountered in Adaptive Information Filtering where the importance of the interplay between *exploitation* and *exploration* has been recognized. We expect that there are various sampling strategies that optimize the balance between exploitation and exploration. By providing our *preliminary* results we illustrate the need and importance of such strategies.

4. Experimental procedure

In order to quantify the effect of relevance feedback and display strategies, we need to define (i) a search task, (ii) an evaluation methodology and (iii) the initial conditions, as discussed in Sections 4.1–4.4. In the experiments we use the Reuters-21578 collection of textual documents. From the documents we extract the contents of two fields, the "Body" and the "Title" and, after removing the stop words, we create a vector representation of documents with *tf-idf* weights. Since some of the documents have empty "Body" fields, we removed them from the collection and arrived at a data set of 19,043 documents.

4.1. Task model

In the context of retrieval, at least three classes of search may be identified (Cox et al., 2000):

(a) *Target document search*—the user's information need is satisfied by a particular document. For example, a researcher may be looking for a specific paper on a research topic.
(b) *Category search*—the user seeks one or more items from a general category or a topic. This task places more emphasis on the content evaluation and often requires subjective relevance judgements.
(c) *Open ended browsing*—the user has some vague idea of what to look for but is open to exploration and may repeatedly change the topic during search.

Of these three scenarios, the target document search, or *known-item search*, is most amenable to evaluation for there are several clear measures of effectiveness. For example, we chose to compare different systems based on the total number of documents presented and examined before the target is found. This number can be compared with the rank of the target document in the initial search, before any relevance feedback is applied. This initial rank represents that number of documents that the user must view and scroll through before reaching the target document. Furthermore, we can focus on a particular aspect of the system by restricting the user's actions, e.g., requiring that the user selects a specified number of documents from the display.

While target document search is typically equated with the 'known item search', the former encompasses a wider spectrum of search scenarios. It can include any information search that is satisfied by a specific document, regardless of whether or not the user is familiar with the target document. So long as the user can recognize that his or her information need is satisfied when a specific document is displayed, we can model that scenario as the target document search.

4.2. Evaluation methodology

In order to examine the effect of relevance feedback and alternative display strategies we devised an experimental procedure that includes the complete space of possible user interactions with the system. More precisely, for a given query or information need, we create *the user decision tree* representing all possible document selections in each feedback iteration. This is feasible because of the small number of documents, D, that are displayed in each iteration. Thus, we can examine all user feedback strategies, including those of an 'ideal user' whose selection of documents minimizes the number of documents that must be examined before retrieving the target document.

In each iteration the tree expands by a factor of D (see Fig. 1), i.e., the number of documents in the individual display. For practical purposes, we limit the number of iterations to five; the initial display of D documents followed by five iterations of relevance feedback. This results in a tree of depth five. For $D = 4$, the maximum number of nodes in the tree is $1 + 4 + 4^2 + 4^3 + 4^4 + 4^5 = 1365$, where a node represents a display of D documents. The tree may be smaller if the target is located earlier since we do not expand the branches of the tree once the target has been displayed. The choice of display size $D = 4$ is motivated by the size of a typical mobile device display. However, the same method could be used to investigate the effect of a range of display sizes.

The *minimum rank* for a given target document corresponds to the case when the user always provides the system with the optimal document for relevance feedback. It is important to note that 'optimal' may not always mean the document most similar to the target.

We also examine the *number of occurrences of the target document* in the decision tree. This enables us to estimate the likelihood that a non-ideal user will locate the target document.

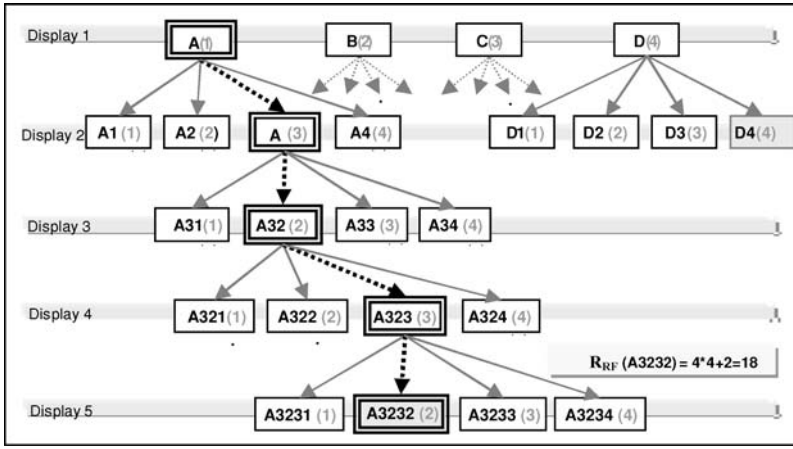

Fig. 1 Decision tree for iterative relevance feedback, showing nodes in which the target document is reached, the rank of a document within each display, and the calculation of RF-rank for the target document labelled A3232

For example, if the target document appears in only one path of the tree, then any deviation of the real user from the relevance feedback of the "ideal" user would result in a failed search. Conversely, if the target document appears in many paths, then the deviations from the "ideal" are still likely to yield successful searches, albeit that these searches require further effort.

We expect that examining sets of documents that are displayed after each iteration will reveal additional properties of the relevance feedback and display strategies. Finally, since the trees are generated automatically, it is possible to create trees for a large number of searches, thereby facilitating a statistical analysis of the algorithms.

4.3. Construction of the user decision trees

Figure 1 provides an example of the user decision tree. At each iteration the tree expands by a factor of $D = 4$. While we are interested in the general behaviour of relevance feedback algorithms, from the application point of view it is most important to understand the impact of the first few iterations of relevance feedback. It is unlikely that the users would engage in a large number of feedback iterations. Therefore we limit the tree expansion to depth five, considering the root of the tree as depth zero.

The initial display of four documents is labelled A-B-C-D and is followed by five iterations of relevance feedback. At each iteration, selection of a document from the display leads to a new branch in the tree. Some branches contain the target document. Since we are focussing on the target document search, we do not expand branches for displays that contain a target document.

We annotate each document in the graph by its rank p within the display of $D = 4$ documents, with p having the value $p = 1, 2, 3,$ or 4. We concatenate displays from relevance feedback iterations by appending to the list the most recent display. The resulting list shows documents in the order in which the user would view them. For each document in the tree, we can identify the corresponding ranked list and calculate the *relevance feedback rank* $R_{RF} = k \cdot D + p$, where k is the number of previous displays, $k = 0, 1, 2, 3, 4$ or 5. R_{RF} essentially corresponds to the number of documents that the user has viewed before locating

the document. In our evaluations we compare R_{RF} with the rank of the document in the initial search. We refer to this baseline rank as the *scroll rank*, R_{Scroll}, since this is the number of documents that the user would have to examine by scrolling down the original list of search results in order to reach the target document.

4.4. Initialisation

We begin experiments by randomly selecting a target document from the database. An initial query is then automatically generated by randomly selecting M terms from the target document. In our experiments $M = 4$. These M terms are used in two ways: as a search query to obtain the baseline search results and as input to the relevance feedback procedure which will further refine the query based on the user's responses. Randomly sampling for query terms does not simulate query generation by users. Rather, it provides us with a method for analysing performance against queries of varying quality—a *good* query is indicated by the target occupying a position high up in the initial ranking, i.e. before relevance feedback is applied. Similarly, a *bad* query is indicated by the target occupying a position low down in the initial ranking. The query vector is simply a vector of equally weighted terms, reflecting our assumption that the user may have some expectations of finding certain terms in the document but is otherwise unaware of the characteristics of the target document or the document corpus in general.

The user is initially shown a display of D documents that are chosen based on which display strategy is being used. The user's response is used by the relevance feedback algorithm to modify the query. The documents in the collection are then scored against the new query and a new display of D documents is presented to the user, based on the search ranking and display strategy. Previously viewed documents are not included in the subsequent search iterations.

5. Results

In our experiments we generated 100 trees, corresponding to 100 distinct target documents, randomly selected from the subset of 19,043 documents from the Reuters collection. The initial query was generated from a sample of terms occurring in the target document and the scroll rank of each target document was recorded.

For each target document we generated a complete search tree based on iterative feedback, with two types of displays: (1) the Top-D display always showing the top 4 ranked documents from the search iteration and (2) the Sampled display that probabilistically selects the documents based on the current ranking of documents in the database. Trees and paths within the trees that contain the target documents are referred to as *successful searches* for the relevance feedback scheme. Tables 1–4 summarize the statistics of the tree displays and successful searches.

6. Discussion

The scroll rank of a target document is the position of the document in the initial ranked list of search results, i.e. the number of documents that the user would have to scroll through in order to reach the target (in the absence of feedback). The RF rank of an *ideal user* is the minimum path length from the root of the tree to a node with the target, whereas the mean

Table 1 Search tree statistics for the three feedback algorithms and two display strategies

	Rocchio feedback algorithm		RSJ Feedback algorithm		Bayesian feedback algorithm	
	Top-D	Sampled	Top-D	Sampled	Top-D	Sampled
Percentage of trees with target	52	97	39	33	52	90
Percentage of paths containing the target	46.67	4.5	27.99	0.087	46.80	4.30
Average R_{Scroll} of targets found in trees	13.79	98.54	37.28	312.03	7.92	64.23
Average min R_{RF} of targets found in trees	6.5	11.25	7.20	17.76	6.13	10.61
Average R_{RF} for the 'average user'	20.53	20.2	20.22	18.26	21.27	19.94

Table 2 Performance of the rocchio RF algorithm based on the initial query

Scroll rank range	Number of targets	Number of targets found		Avg. No. of documents viewed without RF		Avg. No. of documents viewed by the 'ideal user' with RF		No. of docs viewed with RF averaged over successful users	
		Top-D	Sampled	Top-D	Sampled	Top-D	Sampled	Top-D	Sampled
10–20	45	45 (100%)	45 (100%)	4.38	4.38	4.31	5.33	16.54	19.13
21–40	14	6 (42.8%)	14 (100%)	25.5	29.79	20.67	13.07	21.62	21.92
41–60	5	0 (0%)	5 (100%)	–	54.2	–	16.6	–	21.99
61–80	4	0 (0%)	4 (100%)	–	66.5	–	16.5	–	21.80
81–100	6	0 (0%)	6 (100%)	–	92.83	–	15.33	–	21.49
>100	26	1 (3.84%)	23 (89%)	367	341.3	20	18.56	20.78	22.14

Table 3 Performance of the RSJ RF algorithm based on the initial query

Scroll rank range	Number of targets	Number of targets found		Avg. No. of documents viewed without RF		Avg. No. of documents viewed by the 'ideal user' with RF		No. of docs viewed with RF averaged over successful users	
		Top-D	Sampled	Top-D	Sampled	Top-D	Sampled	Top-D	Sampled
1–20	27	27 (100%)	7 (25.9%)	5.67	4.72	4.26	17	19.21	18.67
21–40	6	2 (33.3%)	2 (33.3%)	34	31	7.5	17	12.46	17
41–60	5	3 (60%)	3 (60%)	47.33	41.67	6.33	17.33	7.4	17.33
61–80	8	1 (12.5%)	3 (37.5%)	74	68.33	17	21	18.15	21
81–100	2	1 (50%)	2 (100%)	81	88	24	17	24	17
>100	52	5 (9.6%)	16 (30.7%)	187.2	606	18.2	17.5	21.72	17.94

length of all paths leading to the target represents the average performance of successful users. The first row in Table 1 is the probability that a search (using a given display scheme) will be successful, and row two is the probability that a non-ideal user will find the target. For the Top-D display strategy, about 50% of the trees contain the target (lower for RSJ). In the remaining cases, the target was not found within five rounds of relevance feedback. This

Table 4 Performance of the Bayesian RF algorithm based on the initial query

Scroll rank range	Number of targets	Number of targets found		Avg. No. of documents viewed without RF		Avg. No. of documents viewed by the`ideal user' with RF		No. of docs viewed with RF averaged over successful users	
		Top-D	Sampled	Top-D	Sampled	Top-D	Sampled	Top-D	Sampled
1–20	45	45 (100%)	45 (100%)	4.38	4.38	4.31	5.02	16.54	18.75
21–40	14	6 (42.8%)	14 (100%)	25.17	29.78	17.67	13.07	22.21	21.35
41–60	5	0 (0%)	5 (100%)	–	54.2	–	13.4	–	21.52
61–80	4	1 (25%)	4 (100%)	64	66.5	17	18.5	18.05	21.98
81–100	6	0 (0%)	6 (100%)	–	92.83	–	18.33	–	22.18

percentage is clearly a function of the accuracy of the initial query, which can be judged by examining the scroll rank of the target document. This will be discussed further.

The ideal user represents the best possible performance achievable. Real users are unlikely to perform as well. However, the average number of paths in the tree that contain the target suggests that deviations from the ideal still have a reasonable chance of locating the target document. The average rank of target documents in the tree was obtained by calculating first the average rank for the target document within its particular tree and then averaged over the set of all the trees that contain target documents.

6.1. Top-D display scheme

For the Rocchio and Bayesian algorithms, we see that for a scroll rank of less than 20 (Tables 2 and 4, rows corresponding to scroll rank range 1–20), relevance feedback with Top-D display is successful 100% of the time. For higher values of the initial scroll ranks, i.e.; poor queries, we observe a fall off in the percentage of successful searchers. However, the sampled display approach offers performance that is more constant. For the case of RSJ, with an explicit term expansion strategy, the Top-D display performs better.

6.2. Convergence

It was observed that sub-trees below a node at depth 4 were often identical. That is, the set of four documents displayed to the user at depth 5 was the same, irrespective of the choice of relevant document at the preceding level. Note that the relative *order* of displayed four documents may be affected by the relevance feedback, but the *same* documents appeared in all four sub-trees. It is important to note that the convergence was observed for all three algorithms: even though the sets to which they converged were different.

Since the phenomenon was not symptomatic of any one particular algorithm, we suspect that this *convergence* is due to the greedy nature of the display updating strategy—that of picking the D most probable items (based on the score with respect to the current query). Since the aim of the RF algorithm is to extract *similar* documents from the collection, it results in a situation where successive displays offer no diversity. This could be seen as a direct consequence of the "cluster hypothesis" which states that documents relevant to the same query are likely to be similar to each other. The small variation across the documents in the display is also due to the small number of documents, 4, in the display. However, similar convergence properties were observed for larger displays.

6.3. Sampled display scheme

For the alternative display, a higher percentage of the trees contained the target document with the Rocchio(an increase from 52% for Top-D to 97% for Sampled) and Bayesian schemes (52%–90%, refer Table 1). More importantly, we do not observe a performance degradation as the quality of the initially query degrades. And for very poor initial queries, the alternative display strategy is superior. Since the RSJ algorithm itself considers exploring different regions of the search space by query expansion, use of the sampled display strategy led to an over-adventurous approach, resulting in a smaller number of successful searches and fewer paths leading to the target in a given tree. This illustrates the classical dilemma between exploration and exploitation.

Analysis of the trees containing the target revealed that the average scroll rank was much higher than the rank for an ideal user using relevance feedback and the alternative display, representing a very significant reduction in the number of documents examined. However, once more, we need to recognize that real users are unlikely to perform as well as the ideal user. For the sampled display, the average number of paths in the tree that contain the target is low, which would suggest that deviations from the ideal may have a significant detrimental effect on performance. The number of real users finding the target when using the sampled display, though lower than when using the Top-D display, does not however reflect this expectation (Section 8). This would strengthen the case for the usage of the sampled display update. Finally, we note that the convergence phenomenon observed with the Top-D display was not exhibited using the sampled display.

7. Constructing a statistical model of the "Successful Users"

The simulation-based framework outlined above gave us a method of automatically investigating the effects of every possible user action. Some of these actions were successful (in terms of leading to the target) and most were not. The trees generated provide a data source that can be *mined* to produce a probabilistic model of the "successful users". This is similar to the construction of probabilistic automaton for navigation in hypertext described in Levene and Loizou (1999).

To do this, we construct a Hidden Markov Model (HMM) with H hidden states and O allowed outputs. Here H is the number of displays as dealt with in the trees plus an additional "Found" state. From each of the states corresponding to a display, (O−1) of the allowed outputs can be generated—in our case, these O−1 are each of the possible user actions. The final Oth output is only allowed from the Found state. For our case, $H = 7$ (the initial display, five iterations of feedback and the "Found" state) and $O = 5$ (choose one of four documents or being in the "Found" state). The model is built such that from a given display state, the only allowed transitions are into the next display state, or to the Found state. The diagrammatic representation is provided in Fig. 2.

From the trees that were collected, the sequence of paths representing the choices that led to a successful search were extracted. Ignoring the searches where the target was found in the initial display, the remaining paths were used as training data for the HMM. The trained parameters of the HMM have the following interpretations:

(a) The transition matrix is an estimate of finding a target in a given iteration (a transition to the Found state) against having to move onto the next iteration

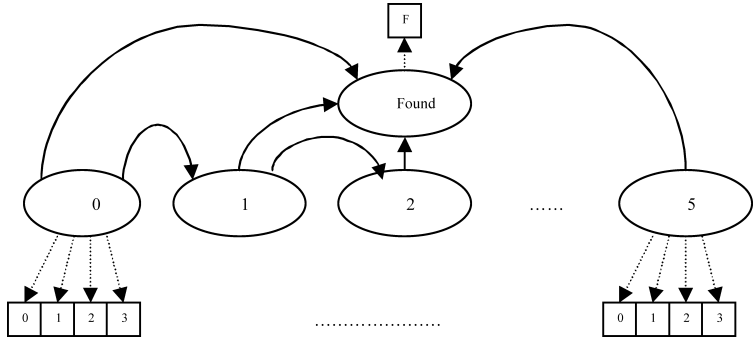

Fig. 2 Tree paths represented as state changes

(b) The emission matrix indicates the optimal choice of 'relevant document' in a given display state.

For each of the 6 variations (3 RF algorithms * 2 display strategies), two sets of trained models are constructed:

(1) Using all successful paths—representing the 'average user'
(2) Using only the shortest path from each tree—representing the 'ideal user'

In the Transition Matrices both the rows and columns correspond to iteration numbers or display states, whereas in the Emission Matrix the rows correspond to the iteration and the columns represent the choice of relevant document in that iteration with the last column being the 'Found' state. It is the Emission Matrix in each case which is of interest. As an example here, we provide the Emission matrices for the model of the 'average user' using the Bayesian feedback algorithm (Table 5).

Table 5 Emission matrices for the trained model for the average user using the Bayesian algorithm

Top-D display				
0.16	0.31	0.28	0.25	0
0.24	0.27	0.21	0.28	0
0.17	0.26	0.28	0.29	0
0.25	0.28	0.23	0.24	0
0.29	0.29	0.23	0.19	0
0.26	0.23	0.23	0.28	0
0	0	0	0	1
Sampled display				
0.26	0.24	0.25	0.25	0
0.26	0.26	0.24	0.24	0
0.26	0.24	0.25	0.25	0
0.28	0.25	0.24	0.23	0
0.33	0.28	0.20	0.19	0
0.56	0.26	0.13	0.05	0
0	0	0	0	1

If in the Emission Matrices of the trained models, the first column dominated every row, this would strengthen the belief in the practice of choosing the highest ranked item in every iteration for feedback, i.e. pseudo-relevance feedback. On the other hand, a uniform distribution across the choice of relevant document (columns 1 to 4 all being 0.25) indicates the absence of any significant pattern. However, we do observe some deviations from both these extremes. For example, in almost all cases, with the Top-D strategy, there seems to be a preference for the lower ranked items (higher values in later columns, indicating the need for 'exploration'). But in the sampled display update scheme, there is a very small bias towards the higher ranked items.

The statistical significance of these matrices is of course open to debate—they are based on only the successful searches of 100 trees built. Plus, it is not clear if they deviate enough from the uniform distribution to warrant being classified as interesting. However, it is another example of how the evaluation methodology can be used to gather other properties which can be used to design the system. A trained HMM is thus the statistical model of all "successful users" across the 100 trees we built. A possible use of such a model would be for pseudo-relevance feedback: in a given state, we can pick which document(s) should be fed back implicitly as being relevant by picking the appropriate columns from the emission matrix with the highest values. Here, we conduct a user trial to validate our user model.

8. User trial

To test if our simulation-based framework corresponds in any way to the behavior of actual users, a small scale user trial of 12 subjects, all of whom were CS/EE PhD students, was conducted.

The user-interface consisted of a screen divided into two sections. The left half, running along the height of the screen, was used to display the 'target' continuously throughout the session. The users were given time to familiarize themselves with this target before proceeding. The right half of the screen was divided into four quadrants, each displaying one of four documents. At each iteration, the user was instructed to indicate the document most relevant to the target by clicking on it. We provided a "Next" button to move to the next display. There was also a progress bar showing the number of completed and remaining iterations. Since most of the subjects were unfamiliar with the specifics of the feedback algorithms, they were not told to base their decisions on textual criteria (i.e. the presence of words) but were free to make their judgment on any basis they deemed useful. A screenshot of the user interface is provided in Fig. 3.

The target and initial display were selected from the simulated user trees in which the target was known to be present in at least one branch of the tree and the target was not present in the initial display. Every user session was thus a walk through one of our previously analysed trees. The trees were constructed on the Reuters-21578 corpus, the articles that were displayed (the targets and the given choices) were all news reports loosely connected to financial matters. Since the topics of such documents were going to be largely unfamiliar to the subjects, the task was made 'interesting' by pointing out to the user that there exists at least one sequence of actions that leads to the target and they had to find one such sequence for each target. This made the trial a sort of 'game', hopefully maintaining user interest throughout the trial.

The trial consisted of each user being given six targets one after the other—corresponding to the 3 RF algorithms and 2 display strategies, the order of which was chosen at random. The results are presented in Table 6. The second column titled "Number Found" gives the

number of users, out of twelve, who found the target for this combination. Each target has a corresponding scroll rank (from the tree) and the "Average Scroll Rank" is the mean scroll rank of the targets chosen to be presented to the user, while the next column provides the scroll ranks of targets that the users found by the interactive process. In each of our successful trees, the target could potentially be present in a number of nodes of the tree. The real users who found the target each trace one of these paths. The average RF rank of these successful searches is given in the fifth column. Time estimates for the successful and unsuccessful users are given in the last two columns.

How do these results compare with our earlier results (Tables 1–4)? It is easy to see that the number of users finding the target using the sampled scheme was less than those using the Top-D scheme. This is to be expected since Table 1 indicates that the percentage of paths containing the target is much lower for the sampled display. In the extreme case, the RSJ algorithm using sampled display had only 0.087% of paths in successful trees leading to the target—none of the real users using this combination found the target. There is also indication of dependence of the success of the user on the time spent—unsuccessful users spent a lesser amount of time on the task.

Comparison of the three algorithms using only the data from the simulations does not reveal a clear winner. However, with the results of the user trial, the Bayesian algorithm with the Sampled Display should be favoured because it not only provides a significant

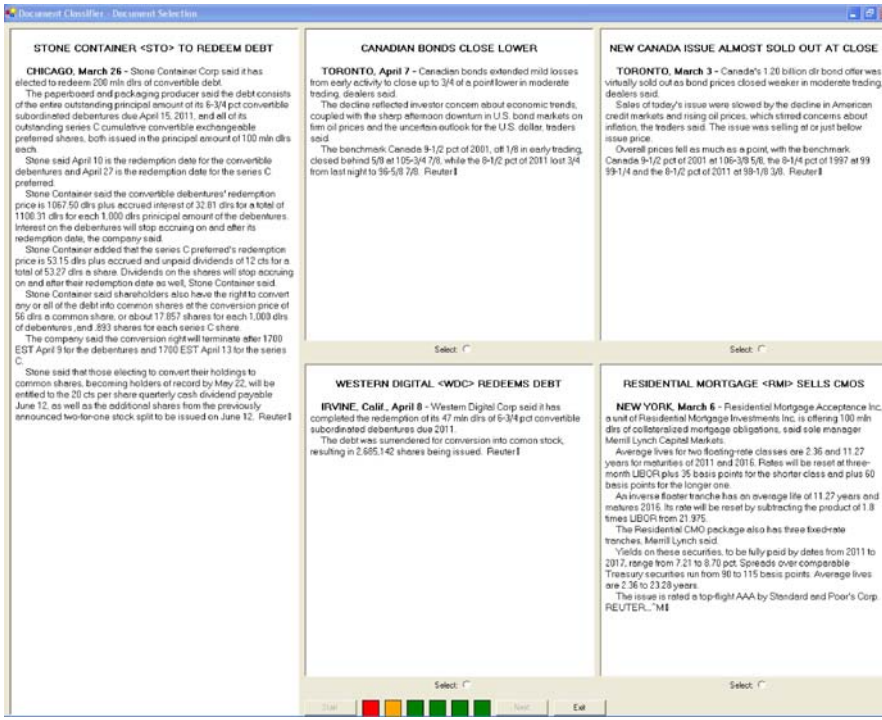

Fig. 3 Screenshot of interface for the user trial. The single window on the left is the target to be found, the four options on the right are the available user choices. The progress bar at the bottom illustrates that this is the second iteration of feedback

Table 6 Summary of user trial results

Algorithm	Number found	Avg. scroll rank	Avg. scroll rank found	Avg. RF rank found	Avg. time for successful (in sec.)	Avg. time for for unsuccessful (in sec.)
Rocchio Top-D	11	47.33	18.27	14.81	162	154
Rocchio sampled	6	34	22.67	11.83	173	135.33
RSJ Top-D	7	107	77.28	12.57	105	258.2
RSJ sampled	0	375	N/A	N/A	N/A	156.58
Bayesian Top-D	11	23.23	19.63	18.09	203.27	295
Bayesian sampled	9	34.42	25.55	11.22	167.55	69

improvement over scrolling (about 50%) but also had a high success rate with the real users (75%).

As described in Section 7, we have 12 trained HMMs—two for each combination of RF algorithm and display strategy. The first HMM was trained on all successful paths in the corresponding trees while the second was trained on the set of shortest paths from each tree. Real users were divided into two subsets—those that were successful (i.e. found the target) for that combination and those that were not. The average probability of the sequence of actions of each action-path in each subset was calculated by following the sequence of actions through the trained HMM.

We then calculate two quantities P1 and P2.

$$P1 = Average\left(\frac{Prob_{ideal}(successful)}{Prob_{average}(successful)}\right) \quad and \quad P2 = Average\left(\frac{Prob_{average}(successful)}{Prob_{average}(unsuccessful)}\right)$$

where $Prob_{ideal}$ is the probability when the path is mapped onto the HMM trained on shortest paths only and $Prob_{average}$ is the probability calculated based on the HMM trained on all successful paths. The results are given in Table 7.

P1 essentially gives an estimate of how close real successful users came to achieving the upper bound as estimated by our simulations. A value higher than 1 for the Bayesian algorithm with the Sampled display means that most users in the trial who found the target did so through the optimal sequence of steps. If we interpret our statistical model as defining a prescribed sequence of actions in order to be successful for a particular algorithm-display update combination, P2 measures the odds of a real user not finding the target despite following the model. The high values here indicate the real unsuccessful users were indeed the ones that did not follow our model. It can of course be argued that since we used the pre-computed

Table 7 Behaviour of real users mapped to the statistical model

Algorithm	*P*1	*P*2
Rocchio/Top-D	0.93	8.84
Rocchio/sampled	0.98	137.79
RSJ/Top-D	0.86	154.03
RSJ/sampled	N/A	N/A
Bayesian/Top-D	0.57	15.28
Bayesian/sampled	1.07	71.18

trees for our user trial, the paths followed by the real successful users would have been actions that were used to train the HMM in the first place. However, the difference in magnitude between the probabilities of the two groups indicates that there are indeed patterns in our HMM which are all the more reliable because we constructed the model after exploiting the complete range of user actions over a large number of trees. This can be verified by removing the paths of the real users from the training set of our HMM, and then calculating the probabilities—the changes in the values were found to be minimal.

9. Conclusions

We examined whether relevance feedback and alternative display strategies can be used to reduce the number of documents that a user of a mobile device with limited display capabilities has to examine before locating a target document. In this scenario, it is possible to construct a tree representing all possible user actions for a small number of feedback iterations. This allows us to determine the performance of an "ideal" user, i.e. no real user can perform better. We are therefore able to establish an upper limit on the performance improvement such systems can deliver. The experimental paradigm has the further advantages of (i) not requiring a real user study, which can be time consuming, and (ii) the ability to simulate very many searches, thereby facilitating statistical analysis.

Using each of three relevance feedback algorithms with a display size of four documents, we constructed 100 trees. With the greedy Top-D display strategy, analysis of the trees containing the target (i.e.; the successful searches) revealed that relevance feedback with Top-D resulted in close to 50% reduction in the number of documents that a user needed to examine compared with simply performing a linear search of a ranked list calculated from the initial query. It should however be noted that this number is exaggerated because of the presence of outliers—the reduction obtained is close to 10% without these cases.

It is unclear as to why the improvement is so low. This may be due to the experimental procedure which required a user to always select one document as relevant, even if none of the displayed documents were actually relevant. Future work is needed to examine whether performance can be improved by: (1) alternative values for the algorithm parameters (2) the identification of non-relevant as well as relevant documents (3) alternative distance metrics.

Similarly, the observation of convergence of the relevance feedback algorithm using the Top-D display also needs investigation. More positively, it was observed that relevance feedback almost never led to worse performance for an ideal user.

We also examined how the performance of the system was affected by an alternative display strategy in which the displayed documents were drawn from the same underlying distribution as the current scores of documents in the database. This sampling strategy crudely approximates a strategy in which we attempt to maximize the immediate information gain from user feedback.

Using this display strategy, the Rocchio algorithm (with no explicit feature selection) and the Bayesian algorithm (which implicitly uses all the features incorporated into the distance metric) had a larger number of successful searches. However, this large improvement may be misleading. The target is present in an extremely small fraction of the 1024 paths of the tree. Thus, while the "ideal" user is guaranteed to find the target, any deviation by real users from the "ideal" is likely to result in a failed search. RSJ's offer weight selection mechanism is known to be unstable, and coupling this with an exploratory display update strategy led to worse performance.

Generalizing, it is clear that if the user's query is sufficiently accurate, then the initial rank of the target document is likely to be high and scrolling or relevance feedback with a greedy display performs almost equally well. However, if the user's initial query is poor, then scrolling is futile and relevance feedback is required—either with a display strategy that explores larger regions of the search space or a feedback algorithm that does the same.

Our simulation-based framework indicated that there is little to choose between the three algorithms considered. But based on the results of the user trial, the Bayesian algorithm coupled with the sampled display update strategy is suggested as being the best. It was however encouraging to note that the predictions made by analyzing the trees corresponded closely to the results of the (admittedly small) user trial.

We also showed a way of capturing all the statistical properties of the trees built in the form of a trained Hidden Markov Model. This HMM is a compact probabilistic representation of all the successful "users" encountered during the tree building. We also showed that the real users who were successful mapped more closely to this trained model than the unsuccessful users.

References

Aalbersberg, I. J. (1992). Incremental relevance feedback. In: *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 11–22). Copenhagen.

Buyukkokten, O., Garcia-Molina, H., Paepcke, A., & Winograd, T. (2000). Power browser: Efficient web browsing for PDAs. In: *Proceedings of the ACM Conference on Computers and Human Interaction*, (CHI'00).

Buyukkokten, O., Garcia-Molina, H., & Paepcke, A. (2001). Seeing the whole in parts: Text summarization for web browsing on handheld devices. In: *Proceedings of the Tenth International World Wide Web Conference (WWW)*.

Campbell, I., & van Rijsbergen, C. J. (1996). The ostensive model of developing information needs. In: P., Ingwersen, & N.O., Pors (Eds.), *Information science: Integration in perspective*, *Proceedings of CoLIS 2*. (pp. 251–268).

Chen, Y., Ma, W.-Y., & Zhang, H.-J. (2003). Detecting web page structure for adaptive viewing on small form factor devices. In: *Proceedings of the Twelfth World Wide Web Conference*. Budapest.

Cox, I. J., Miller, M. L., Minka, T. P., Papathomas, T. V., & Yianilos, P. N. (2000). The Bayesian image retrieval system, PicHunter: Theory, implementation and psychophysical experiments. *IEEE Transactions on Image Processing*, *9*(1):20–37.

Dunlop, M. D. (1997). The effect of accessing non-matching documents on relevance feedback. *ACM Transactions on Information Systems*, *15*(2):137–153.

Harman, D. (1992). Relevance feedback revisited. In: *Proceedings of 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Copenhagen, 1.10.

Jones, M., Marsden, G., Mohd-Nasir, N., Boone, K., & Buchanan, G. (1999). Improving web interaction on small displays. In: *Proceedings of the 8th World Wide Web Conference*. Toronto, Canada.

Jones, M., & Marsden, G. (1997). From the large screen to the small screen-. Retaining the designer's design for effective user interaction. In: *IEEE Colloquium on Issues for Networked Interpersonal Communicators, 239*(3):1–4.

Levene, M., & Loizou, G. (1999). A probabilistic approach to navigation in Hypertext. *Information Sciences, 114*:165–186.

Magennis, M., & van Rijsbergen, C. J. (1997). The potential and actual effectiveness of interactive query expansion. In: *Proceedings of 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Philadelphia.

Harman, (Ed.) (1997). Overview of the fifth text retrieval conference (TREC-5). Gaithersburg, MD: NIST.

Rocchio, J. (1971). Relevance feedback information retrieval. In: Gerard Salton (Ed.), *The smart retrieval system—experiments in automatic document processing* (pp. 313–323). Prentice-Hall, Englewood Cliffs, N.J.

Robertson, S. E., & Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, *27*, 129–146.

Rodden, K., Milic-Frayling, N., Sommerer, R., & Blackwell, A. (2003). Effective web searching on mobile devices. In: *Proceedings of the HCI Conference*, Bath.

Ruvini, J.-D. (2003). Adapting to the user's internet search strategy. IUI'03, Miami, Florida.

Sellen, A. J., Murphy, R., & Shaw, K. L. (2002). How knowledge workers use the web. In: D., Wixon (Ed.), *Proceedings of CHI 2002, ACM* (pp. 227–234).

White, R. W., Jose, J. M., van Rijsbergen Cornelis, J., & Ruthven, I. (2004). A simulated study of implicit feedback models. In: *Proceedings of ECIR*.